状態遷移テストにおけるカバレッジ基準の提案とモデル検査を用いたテストケース生成について

土屋達弘[†1]　カッシア ジ ソウザ カルヴァーリョ[†1]

状態遷移テストについて，新しいカバレッジ基準を提案する．具体的には，このカバレッジでは，仕様上の二つの状態 s, s' からなる各順序対 (s, s') について，それらを順に通るパスをテストすることを求める．また，遷移の順序対についても同様にカバレッジ基準を設定できる．更に，モデル検査を用いて，これらのカバレッジ基準を満たすテストケースを生成できることを示す．

New Coverage Criteria for State Transition Testing and Model Checker-Based Test Case Generation

Tatsuhiro Tsuchiya[†1] and Cassia de Souza Carvalho[†1]

New coverage criteria are proposed for state transition testing. A new coverage criteria requires that for every ordered pair (s, s') of two states in the specification of the SUT, a path that goes through s and then reaches s' be exercised. A similar criteria can be naturally defined for ordered pairs of transitions. Model checker-based test case generation is also touched on.

1. はじめに

仕様に基づくテスト手法として，状態遷移テストがよく知られている．この手法では，状態遷移が表現された仕様が利用できることを前提に，仕様上の状態すべてに到達するようにテストを行う．同様に，仕様上のすべての遷移を実行するように，テストを行う場合もある．以降，これら 2 種類の典型的なカバレッジ基準を，それぞれ，全状態カバレッジ，全遷移カバレッジとよぶ．

本研究では，新しいカバレッジ基準として，全状態ペアカバレッジ，全遷移ペアカバレッジを提案する．また，モデル検査を用いて，これらの新しいカバレッジ基準を満たすテストケースが生成できることを示す．

2. 提案するカバレッジ基準

状態遷移を記述した仕様が与えらえることを仮定する．このような仕様としては，たとえば，ステートマシン図がある．仕様上の状態の集合を S とする．ここで S は状態の種類（たとえば，ステートマシン図であれば，AND 状態，OR 状態など）に関係なく，すべての状態とする．全状態ペアカバレッジを次のように定義する．

> 全状態ペアカバレッジが満たされるのは，任意の順序対 (s, s'), $s, s' \in S$ について，初期状態から s に到達しその後 $\star 1$ s' に到達する動作が存在しないか，その動作がテストされた場合である．ただし，$s = s'$ の場合も含む．

遷移に関しても同様に，遷移集合に対してその要素の順序対すべてをテストした場合に満たされるカバレッジ基準を定義できる．これを全遷移ペアカバレッジとする．

通常の状態遷移テストでは，全状態カバレッジ，および，全遷移カバレッジがテスト基準として用いられる．全状態カバレッジは，すべての状態に到達することを求める基準である．定義より，全状態ペアカバレッジは全状態カバレッジを包摂 (subsume) する．同様に，全遷移カバレッジは，すべての遷移を実行することを求める基準であるので，全遷移ペアカバレッジは全遷移カバレッジを包摂する．

3. モデル検査を用いたテストケース生成

多くのモデル検査器では，設計が時相論理仕様を満たさない場合，反例を出力する機能を備えている．こ

†1 大阪大学
Osaka University

$\star 1$ s に到達した時点も含む．

の機能を利用して，テストケースを生成する手法がよく研究されている[1)–3)]．また，ステートマシン図をはじめとした状態遷移仕様をモデル検査器の入力言語に変換する研究も，盛んに行われている[4)–6)]．そこで，状態遷移仕様を表現したモデル検査器へ入力できるプログラムが利用可能であることを想定できる．

ここでは具体的にモデル検査器 NuSMV への入力プログラムとして，ステートマシン図が表現されていることを想定する．具体的には，文献[4)] の手法でプログラムが得られているものとする．この手法では，状態を以下の様な変数で表す．

```
VAR
  Alt-Layer: {High, Mid, Low}
  Alarm: {Shutdown, Operating}
```

Alt-Layer, Alarm はそれぞれ状態で，その状態の下位の状態として，それぞれ High, Mid, Low という 3 状態と，Shutdown, Operating の 2 状態があることを表す．

また，各状態について，現在その状態にシステムがあるかどうかを表すブール値をマクロにより定義する．

```
DEFINE
  in-High := in-Alt-Layer & Alt-Layer = high;
  in-Operating
    := in-Alarm & Alarm = Operating;
```

以下，テストケースを生成するために用いる時相論理仕様を，状態対に関して示す．例として，状態対 (High, Low) を考える．CTL を時相論理として用いた場合，以下の時相論理仕様（時相論理式）を用いる．

$$\neg EF(in\text{-}High \wedge EF in\text{-}Low)$$

これは NuSMV の入力言語では以下のように表現できる．

```
CTLSPEC
  !EF (in-High & EF in-low)
```

式の意味は以下の通りである．否定記号を除く部分は，ある条件を満たす大域状態への動作が初期大域状態から存在することを表している．この条件とは，その大域状態が仕様上の状態 High に対応しており，かつ，その時点から仕様上の状態 Low に対応する大域状態へ到達する動作が存在するというものである．否定記号により，そのような動作が存在する場合，この時相論理式は偽と評価され，反例が出力される．NuSMV によってこうして得られる反例は，初期状態から両状態を順に通る動作になっており，その動作をテストケースとすることができる．

上記の手順は，一つの状態対についてのものであるが，これを全状態対について実行することで，全状態ペアカバレッジを満たすテストケース集合が構成できる．また，全遷移ペアカバレッジについても，同様にしてテストケース集合を生成できる．

4. おわりに

本研究では，状態遷移テストに対して，状態対，および，遷移対を網羅する新しいテストカバレッジ基準を提案した．更に，モデル検査器を用いて，提案したカバレッジ基準を満たすテストケース集合を生成する手法を提案した．

今後の課題としては，提案カバレッジ基準の有用性の評価，テストケース生成の自動化，冗長なテストケースの削減などが挙げられる．

参考文献

1) Hong, H., Lee, I., Sokolsky, O. and Cha, S.: Automatic Test Generation from Statecharts Using Model Checking, *Proc. of First Workshop on Formal Approaches to Testing of Software (FATES '01)*, Aalborg, Denmark, pp.15–30 (2001).
2) Lindstrom, B., Pettersson, P. and Offutt, J.: Generating Trace-Sets for Model-based Testing, *Proc. 18th IEEE International Symposium on Software Reliability (ISSRE '07)*, IEEE Computer Society, pp.171–180 (2007).
3) Hamon, H., de Moura, L. and Rushby, J.: Generating Efficient Test Sets with a Model Checker, *Proc. of 2nd International Conference on Software Engineering and Formal Methods (SEFM '04)*, IEEE Computer Society, pp.261–270 (2004).
4) Chan, W., Anderson, R.J., Beame, P., Burns, S., Modugno, F., Notkin, D. and Reese, J.D.: Model Checking Large Software Specifications, *IEEE Trans. on Software Engineering*, Vol.24, No.7, pp.498–520 (1998).
5) Chan, W., Anderson, R.J., Beame, P., Jones, D.H., Notkin, D. and Warner, W.E.: Optimizing Symbolic Model Checking for Statecharts, *IEEE Trans. on Software Engineering*, Vol.27, No.2, pp.170–190 (2001).
6) Zhao, Q. and Krogh, B.H.: Formal Verification of Statecharts Using Finite-State Model Checkers, *IEEE Trans. Control Systems Technology*, Vol.14, No.5, pp.943–950 (2006).